and wherein said processor selectively draws said [open] window object in an overlapping relationship with said other window object based upon said position indicating information in said data structure.

REMARKS

Reconsideration and allowance of the above-identified application are respectfully requested. Claims 1-7 are currently pending.

Applicant acknowledges receipt of the Substitute PTO-948 form. Formal drawings will be submitted after this application has passed to issue.

Examiner objects to the drawings because Figures 1-4 are not labeled as prior art. Applicants are filing concurrently with this paper a Request for Approval of Drawing Changes which requests the required changes. Approval of the drawings is respectfully requested.

Claims 1-7 were rejected under 35 U.S.C. § 112, second paragraph, as allegedly being indefinite for failing to particularly point out and distinctly claim the subject matter set forth therein. By way of the foregoing amendments, each of the concerns raised by the Examiner have been addressed, and Applicant accordingly requests the Examiner to reconsider and withdraw this rejection.

Claims 1-7 were rejected under 35 U.S.C. § 103 as allegedly being obvious over Bates et al. Applicant respectfully traverses this rejection for the following reasons.

According to exemplary aspects, the present invention provides a graphical user interface (GUI) which generates a list indicating the order of windows displayed on a first portion of a display screen. In one exemplary embodiment, the windows may be displayed as a stack of cascaded windows as depicted in Figure 2. When a window is removed from the stack, it may be displayed in a second portion of the display screen (as depicted as (14) in Figure 1). When the user is finished using the removed window, he provides a command to the graphical user interface, whereupon the graphical user interface will replace the removed window to its original location in the stack by making reference to the list. In contrast, prior art graphical interfaces failed to return a replaced window to its original position.

Bates et al. fails to teach or fairly suggest Applicant's invention as characterized above. Admittedly, Bates teaches means for displaying a series of windows in a specific order (as illustrated in Figure 2C). The order of windows may be displayed as a window list (as depicted as feature (24) of Figure 2D). However, the order of windows in the list is a dynamic variable computed on the basis of the extent to which the user has used the window in the past (col. 4, lines 15-26). For instance, within the context of the cascaded display of Figure 2C, the window on the top of the stack is the most-frequently used window, whereas the window on the bottom of the stack is the least-frequently used window. Thus, if a user removes a window from the stack of cascaded windows shown in Figure 2C, there is no guarantee that it will be returned to its original position in the stack when the user is finished with the window. For instance, suppose the user pulls the least-frequently used window from the bottom of the stack. If the user works for a

significant amount of time on the window, by the time the user is finished, the window may have been reclassified as the most-frequently used window. Accordingly, the window will be returned to the stack on the top, rather than at is original bottom position. Bates et al. therefore most clearly teaches away from the present invention.

This distinction is reflected in the claims. For instance, claim 1 recites *inter alia* a step of "generating a list which provides a front-to-back order of said plurality of cascaded windows and an indicator of whether each of said plurality of cascaded windows is currently in its respective original, cascaded position". When it comes time to return a window to its original position, the method includes a step of "returning said window to said original position based upon said list generated by said step of generating". In marked contrast, Bates generates a list which is a function of the activity level of each window (see steps 525-550 of Figure 9A). The list does not include "an indicator of whether each of said plurality of cascaded windows is currently in its respective original, cascaded position" as claimed. Also, when the user is finished with a window in Bates, this window is not returned to its original position using *the same list as previously generated*. Rather, Bates recomputes a new list which factors in the new activity level of the windows, and displays the cascaded windows according to the new list (not the previous list). As mentioned, Bates technique may have the end result of returning a displaced window to a position different from its original position. As a matter of fact, Bates actually discourages the technique of returning a displaced window to its original position. Bates objective is to display windows according to their level of use by the computer operator.

Claim 2 depends on claim 1, and therefore incorporates the allowable distinctions of claim 1 pointed out above. In addition, claim 2 elaborates on the step of returning a displaced window by reciting an iterative procedure whereby an identifier of said window is compared with an identifier associated with each window in said list until a match occurs, and then placing said window behind a window which is next in order in said list after said match occurs. Bates orders windows for display using the activity level of the windows according to the algorithm depicted in steps 525-550 of Figure 9A. Bates *inter alia* does not search a list for an identifier corresponding to a window to be returned to its original position.

Claim 3 depends on claims 2, and in turn claim 1. Claim 3 therefore incorporates the allowable features addressed above with respect to claim 1 and 2. Furthermore, claim 3 recites a step of placing said window behind said window which is next in order in said list after said match occurs only if said window is currently in its respective original, cascaded position. Bates fails to even consider the possibility that more than one window may be out of place from the cascaded stack of windows, and accordingly fails to met the additional features of claim 3.

The remainder of the claims, 4-7, include many of the same features identified above with respect to claims 1-3. Accordingly, these claims are allowable for at least the same reasons proffered above with respect to claims 1-3. For instance, claim 4 recites "placing said first window behind said second window in said first portion of said display space *by making reference to said list*" (emphasis added). Claim 6 recites maintaining a data structure which provides a time-invariant order of window objects, and a processor

- 8 -

"for drawing said window object on said display at said original position using said information in said data structure".
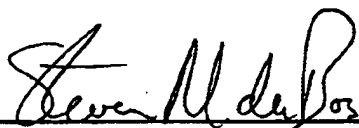
For at least the reasons set forth above, it is respectfully submitted that Applicant's claims 1-7 are not taught or fairly suggested by Bates et al. Accordingly, it is respectfully requested that the rejection of these claims under 35 U.S.C. § 103 be reconsidered and withdrawn.

All of the objections and rejections raised in the Office Action having been addressed, it is respectfully submitted that the present application is in condition for allowance and a notice to that effect is earnestly solicited. Should the Examiner have any questions regarding this response or the application in general, he is urged to contact the undersigned to expeditiously resolve the outstanding issues at (703) 838-6642.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

By: _____

Steven M. du Bois
Registration No. 35,023

Post Office Box 1404
Alexandria, Virginia 22313-1404
(703) 836-6620

Date: _____April 18, 1996_____

- 9 -